AFRL-RI-RS-TR-2017-143

NEON VISUALIZATION ENVIRONMENT

NEXT CENTURY CORPORATION

JULY 2017

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2017-143 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
PETER A. JEDRYSIK
Work Unit Manager

/ S /
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
JULY 2017	FINAL TECHNICAL REPORT	SEP 2012 – MAR 2017

4. TITLE AND SUBTITLE
NEON VISUALIZATION ENVIRONMENT

5a. CONTRACT NUMBER FA8750-12-C-0336
5b. GRANT NUMBER N/A
5c. PROGRAM ELEMENT NUMBER 62702E

6. AUTHOR(S)
Tonya Cannariato, Tom Panning, Larry Mize

5d. PROJECT NUMBER XDAT
5e. TASK NUMBER A0
5f. WORK UNIT NUMBER 19

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Next Century Corporation
3120 Fairview Park Drive, Suite 700
Falls Church, VA, 22040

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/RISB
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RI
11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2017-143

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Review of Next Century Corporation's participation in the XDATA program, and the progress toward the final state of its software deliverable, Neon and Neon GTD.

15. SUBJECT TERMS
Big data, big data analytics, big data visualization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**PETER A. JEDRYSIK**
U	U	U	UU	35	**19b. TELEPHONE NUMBER** *(Include area code)* **(315) 330-2150**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. PROGRAM RESEARCH OBJECTIVES AND TECHNICAL APPROACH SUMMARY

Neon aimed to solve the problem of characterization and visualization of large data sets in ways that allow users to gain rapid insights. Neon supported the DARPA XDATA program by developing computational techniques and software tools for analyzing and visualizing large volumes of data, both semi-structured (e.g. tabular, relational, categorical, metadata) and unstructured (e.g. text documents, message traffic). The objective of this effort was to design and develop a framework for interactive data exploration and easy integration amongst disparate visualizations. This framework is agnostic to the data, so that integrated visualizations and analytics applications can be applied in multiple contexts.

Neon impacted XDATA by affording both visualization developers and users unprecedented automation and flexibility. Furthermore, XDATA was able to quickly release new Widgets by which users could interactively and collaboratively explore data and compose analytic workflows for the scientific discovery and understanding of unknown activities and associated trends, patterns, and relationships.

The overall technical strategy for Neon was to develop a lightweight, presentation-tier architecture that allowed XDATA users to visualize and analyze data from their Web browsers. Neon consists of two fundamental components: the Widget Development Framework (WDF) and the Widget Interaction Framework (WIF). The WDF allows the user to characterize data sets and compose visualization solutions tailored to the data sets. The WIF provides a library of visualization widgets that can be composed within WDF.

2. INTRODUCTION

Neon is a novel departure from conventional information visualization: the goal was not to develop a typical visualization *toolkit* but instead a visualization *environment*. The Neon approach began with the critical insight that the fundamental design, development, and deployment principles of contemporary visualization toolkits are inconsistent with the XDATA vision because they regard the process of visualizing information as a *pipeline*. The Visualization Toolkit (VTK - http://www.vtk.org/), which is the most popular visualization system in use today, explicitly invokes the pipeline metaphor to describe the steps leading from data to a rendered image. Figure 1 shows a simplified version of the pipeline concept: the visualization developer examines the data source, imports the graphics code from a library to model the data, codes the user interface components, compiles the code, and executes the code in the visualization program. Most visualization systems follow this pattern, which entailed numerous disadvantages for XDATA. For example, each user interface will be an inherently “stovepiped” application with embedded interface controls unsuitable for collaborative query, multiple looks at data, analytical workflows, rapid customization, and so on. The rendering of the visualization also requires that the toolkit software is running, to the detriment of analysts who cannot install on their local machines because of security or perhaps licensing restrictions.

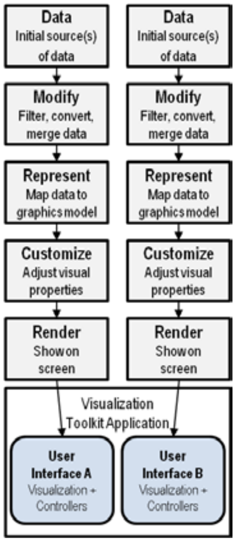

Figure 1: Pipeline approach of conventional visualization toolkits.

Neon re-imagined the process of visualizing information as an *ecosystem*, where display, query, and control elements of the environment adapt through interaction with each other and end users. Figure 2 illustrates the concept. In the Development Framework, the developer would use recommendation tools to analyze the data source and point to a graphical representation from its library that can model the data. The developer would then import the recommended graphics code, customize the interfaces of the graphical model with the data, and use another recommendation tool to analyze the visual salience of the rendered product. Once complete, the visualization would be published to the Interaction Framework, where it is available to anyone with a Web browser. Analysts are then able to run multiple visualizations (“Widgets” in Neon parlance), change them simultaneously with universal controllers, and create analytic workflows for successive query refinement. Importantly, evaluators can develop detailed user models by collecting data about browser activity. The developer tools in the Development Framework incorporate the user models to make more accurate recommendations, improving the next generation of visualizations.

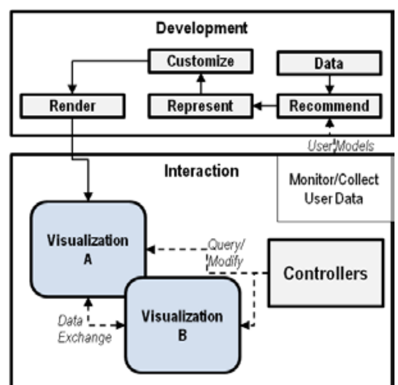

Figure 2: Ecosystem approach of the Neon Visualization Environment.

Neon’s revolutionary ecosystem model for information visualization strongly advanced the priorities of the XDATA program. Table 1 summarizes our technical approach for implementing the model as aligned with the goals of XDATA TA2. We also created a set of goals for Neon that quantified our expected performance for each of the XDATA TA2 goals.

Table 1: Neon technical approach and goals are aligned with XDATA.

XDATA TA2 Goal	Neon Approach	Neon Goal
Develop visualization principles and implementations scalable to data volume and distributed computer architectures.	Develop Widget Interaction Framework as a thin-client system. Neon server will host all information visualization and analysis Widgets and interface with XDATA TA3 cloud services. Users will be able to access XDATA on any system with a Web browser. Apply principles from Cognitive Fit Theory to the filtering and aggregation of data at multiple scales. Employ a bottom-up visual attention neuromorphic model to help Widget developers enhance the areas of high information content with areas of high visual salience.	95% of Widgets rated in highest TA4 evaluation category.
Minimize design-to-execution/testing of a new visualization, with principles to innovate rapidly.	Build a Widget Development Framework with standard interfaces for data visualization, alerting, filtering, and scaling for rapid insertion into Widget Interaction Framework. Create automated assistant tools within WDF that recommend visualization types based on statistical characterization of the source data (Broker) and predicts the regions of the visualization that analysts will experience as visually interesting (Attention Map).	Median development time for new visualization 10x faster than conventional methods.
Incorporate dynamically changing collaborative layouts.	Build Neon on the foundation of the Ozone Widget Framework, a browser-based presentation tier architecture. OWF's flexible customization features allow users to select applications (Widgets), arrange their layout, set up multiple dashboards, and create analytical workflows. Implement the peer-to-peer protocol and Operational Transformation methods to enable collaboration between multiple users on a single Neon display.	Support up to 25 simultaneous users on a single Widget dashboard.

In January 2013, the visualization task performers, USC Institute for Creative Technologies, Parsons Institute for Information Mapping (PIIM), Oculus Info, Kitware, Continuum, MDA/JPL/USC, and Next Century met in a working session to identify each performer's unique strengths, determine common capabilities and methods to work together, and identify early, specific "mini projects" to start working on. The visualization process was mapped across three needs: User Interface Design and Implementation, Data Management, and Visualization Specification and Grammars. These were mapped to the various performers, and the Next Century team placed itself across the Design and Data Management quadrant. As such, as of the March 2013 program review, Neon had become the principal TA2 interface with TA1 analytic components, able to demonstrate tabular and geospatial widgets to show different views of the CharityNet data set. By the time of the July 2013 Summer Camp close-out reports, there was little further conversation about the TA3 or TA4 providers, and the program shifted to focus on the data and how they were visualized as opposed to research software integration and evaluation.

Neon impacted XDATA by affording both visualization developers and users unprecedented automation and flexibility. Furthermore, Neon was able to quickly release new Widgets by which users can interactively and collaboratively explore data and compose analytic workflows for the scientific discovery and understanding of unknown activities and associated trends, patterns, and relationships.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

The Neon server provides filter management and database abstraction. By using the Neon server, client applications can focus on developing visualizations without being tied to a particular database technology or working on the standard interactions between visualizations. The Neon filter management means that any visualizations in a Neon application will automatically work on a consistent, shared subset of the data while still being able to present unique, separate views of that data.

The Neon server is written in Groovy (a Java VM language), but can be accessed by any tool over the HTTP REST interface. A JavaScript library for accessing the REST interface is included to further ease the development of web applications. This interface allows common database querying operations including, selecting, grouping, and aggregating data. It also allows for querying metadata attributes and geospatial queries. All of these operations are provided through a database-independent API that allows the same application code to be used no matter what database houses the data.

Neon's support for database technologies has evolved as the technologies behind the big data movement have evolved. The first implementation supported Hive, a Hadoop technology for interacting with large tables. Hive supported structured queries, but was unable to provide results at interactive speeds. Shark was the first attempt to fix this problem by keeping the tables in memory, and Neon added support for Shark as it was developed. Eventually Shark was subsumed into the growing Spark project, and Neon tracked these changes as Spark began providing acceptable performance. Spark eventually grew sufficiently that we could explore real-time heat maps as well as abstract rendering using that tool.

While Spark and its ancestors can scale to very large sizes and eventually produced good performance for large data sets, they do not scale down very well. Even the simplest, smallest installation is quite involved, and does not perform particularly well for small data sets. For dealing with smaller data sets, and to provide a way for developers and users to experiment with Neon with minimal investment, Neon added MongoDB support. MongoDB is simple to install, configure, and populate, and provides quick responses for smaller data sets.

In the course of evaluating other database technologies for suitability to the needs of Neon applications, we discovered that Elasticsearch had added enough aggregation capabilities to support Neon operations. After testing Elasticsearch performance on common Neon queries, we determined that in many ways it offered performance comparable to the best of MongoDB and Spark. At small scales, Elasticsearch is easy to install and configure and provides very quick responses. At large scales, Elasticsearch is no more difficult to work with than Spark, and is able to roughly match its performance. We added Elasticsearch 1.x support, and as Elasticsearch 2.x became stable and preferred by other teams, that support was upgraded.

Other database technologies we explored through the course of the program included: presto.io; Mondrian; Nanocubes; Datavore; and inMems. One of the more fruitful of these database explorations includes contact with Kai Zeng (UC Berkeley) and Sameer Agarwal (Databricks) for early access to the BlinkDB/G-OLA preview code. In this instance, rather than build atop the deprecated BlinkDB release, we worked with the new Spark package under development in this environment. This process shows the user interface connecting to online aggregation, demonstrating decreasing errors over time as the query process continues. This helped the team explore options for server push of data to our user interface, which we demonstrated using SocketIO.

3.1 Server-Side Transformations

Originally, Neon was meant to be mainly a server and framework that provided a way for developers to generate data transformation tools that would be useful to end users. Its origin in OWF widgets and the simple demonstration of the JavaScript API quickly evolved into an application in its own right. By the end of the 2013 Summer Camp, the Ozone Widget Framework was being relied on for technology-agnostic composition of web browser-based applications in a common display environment, allowing applications to communicate with each other across domains. At the same time, the Neon Framework had emerged in three components to complement, but not retain dependency on, OWF. The Neon Server contained REST services connected to big data stores to perform querying, filtering, and aggregation. The Neon Client was a Javascript library from which visualizations could invoke Neon services. Neon Widgets are visualizations that can be applied to multiple data sets, interactively. The first evolution of the technical approach can be seen in figure 3.

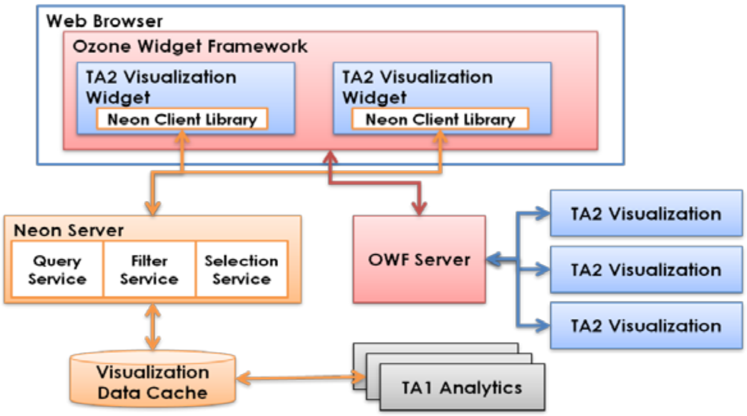

Figure 3: Neon architecture v1.

Half a year later, in the November 2014 plan, the architecture had evolved to reflect something much closer to its final state, moving away from OWF, and incorporating the Neon server element to retrieve and filter data, as well as the visualization widgets that would draw on elements stored in the Neon client library (Figure 4).

The final iteration of the Neon architecture unified the Neon Client library and Neon Server into the Neon Core Project, and added the GeoTemporal Dashboard as the visualization layer on top of the data interaction and retrieval layer (Figure 5).

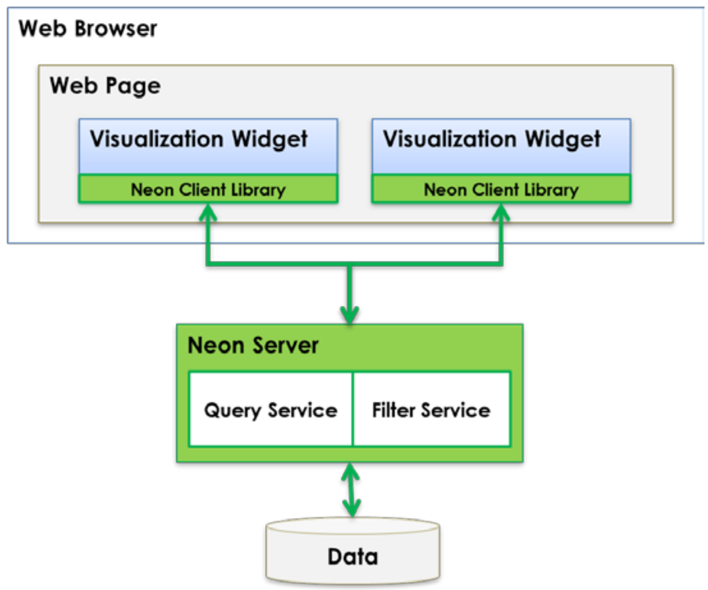

Figure 4: Neon architecture v2.

Ultimately, the result of these many experiments resulted in a tool that allows for database abstractions. The range of database connections available for end users is both broad and unusual in the commercial ecosystem. It allows for common concepts like select, group, and aggregate, as well as geospatial querying. The metadata system allows applications to receive basic field types in the dataset in a database-independent fashion.

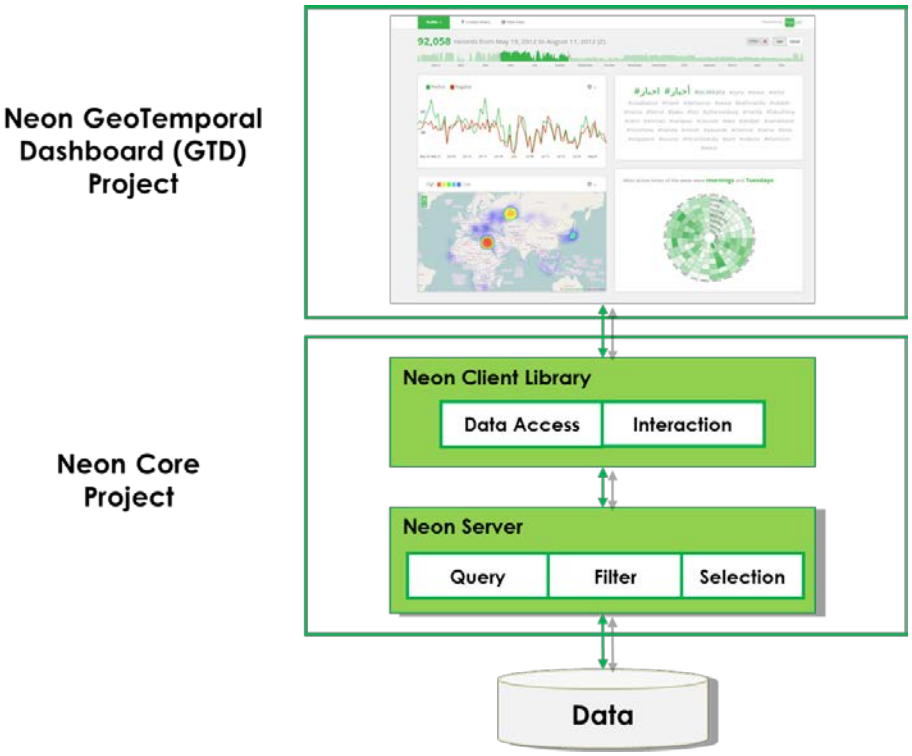

Figure 5: Final Neon architecture

4. RESULTS AND DISCUSSION

The original version of the Neon Dashboard was written in a single page AngularJS 1 application. This allows user to save and load configurations for future use or sharing with other users. It could be fully configured server side, or could be run with just the default configuration, allowing users to create dashboards on an ad hoc basis. The widgets can read the filter state so tools that do not use Neon can still be integrated to provide utility for end users. The tool also allows import and export to and from common formats such as CSV and Microsoft Excel. It also allows for on-the-fly text translation, translating what the user is viewing, and caching the results for future reference.

At the end of the 2013 Summer Camp, the tool was already able to discover and visualize patterns temporally, graphically, and numerically in real time; filter out noise; and ask questions about data points of interest. During that Summer Camp exercise, the tool proved itself on the strength of having integrated eight visualizations and applying them to four datasets. The experiments in those datasets had the following results: In the Kiva dataset, we were able to recognize the characteristics of those who default on high-value loans. In the Bitcoin dataset, we were able to quickly extract the largest transaction values as well as the users with the highest number of transactions. In the Akamai dataset, we could distinguish cultural differences in Internet usage patterns. And in the Twitter dataset, we were able to discover life patterns among the most active Twitter users.

As a result of the 2014 Summer Workshop, Neon was able to incorporate Draper usability testing, and integrate with Tangelo and Aperture Tiles as part of SOCAF and related efforts. Next Century was also able to use OpenCPU to demonstrate R integration as well as call Giant Oak's MMPP and Purdue's STL2 Time Series Analysis. Based on feedback from the User Testing, we moved to an augmented dashboard with a composable, grid-based view that allows users to mold the interface to fit their current workflow or questions.

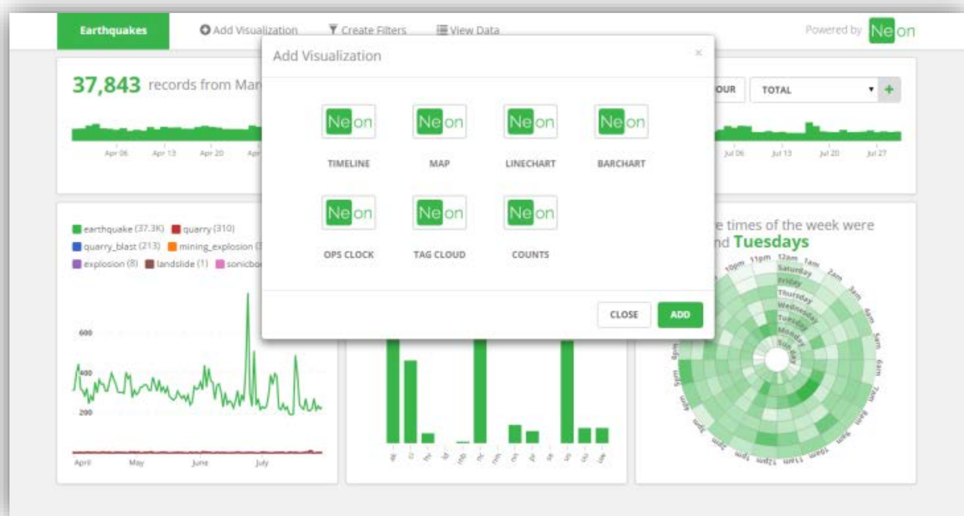

Figure 6: Neon Widget selection interface

4.1 Promoting Developer Adoption

One of the main goals of this program is to advance the state of the art in data visualization. As such, it comprises fundamental research. One of the early directives of the program was to make the code base open source and encourage public contribution to the code base. We do this by publicly hosting our repository on Github as well as maintaining a public demo server at http://demo.neonframework.org/. We provide API documentation, a User manual, and installation documentation to allow developers unaffiliated with the XDATA research program to easily follow our methods and processes to replicate our results. We share a public version of our Docker image at https://hub.docker.com/u/nextcentury/dashboard/, which includes its own instructions on interacting with and installing the quick-start container for the Neon Framework. We also publish similar information on NPM (https://www.npmjs.com/), the package manager for JavaScript.

4.1.1 Collaborations

Following through with the directive toward integrating XDATA tools across a wide range of developers, the Neon team supported collaborative efforts with a plethora of users and organizations. The most fruitful of these has been the work with Charles Stark Draper Laboratory. We integrated Draper's Activity Logger library to participate in Draper scenarios and A/B user testing. We began working with them for user testing in preparation for the Summer Workshop in June 2014, and continued developing that relationship until the end of the program. Datasets involved in these tests included Tweets from South America as well as Tweets from New York City taxis. This work involved creating two separate Amazon images, which have different dashboard contents and layouts to allow Draper to pursue tests of different hypotheses. It also requires a REST service to provide data to the Draper ATAK device.

The South America dataset was a database of 15 million tweets. It was used as a benchmark to test Neon's scalability and required multiple aggregation queries on all records and subsets. The results of these benchmarks are reported in section 8. The New York City taxi dataset was part of a Neon scalability stress test involving a live demo with 45 million rows of data, which then grew to 165 million rows. We learned that response time depends almost entirely on database speed.

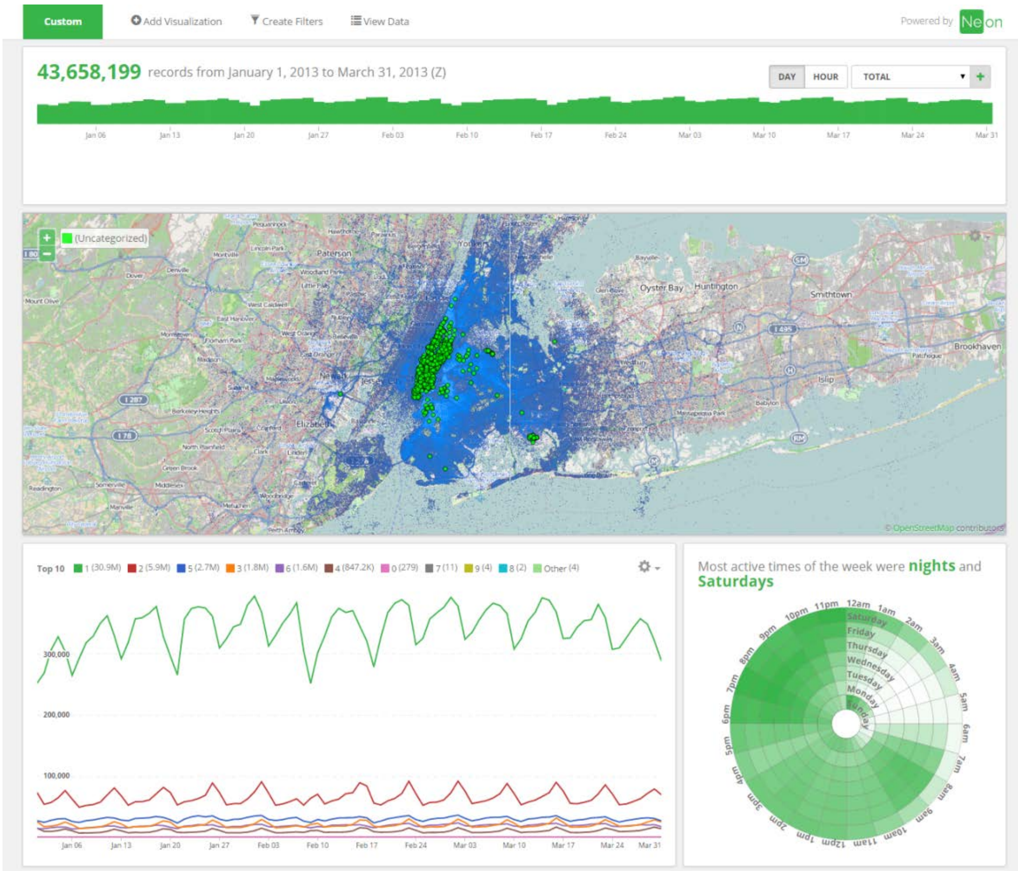

Figure 7: New York City taxi data visualization

Ryan Hafen, while he was collaborating with Purdue University, helped direct us in integrating Time Series analysis and other R-based analyses. These were all integrated via the OpenCPU library. This branch remains on Github to allow further experimentation. We also integrated Giant Oak's R-based Markov Modulated Poisson Process (MMPP) time-series analysis in collaboration with their researcher, Graham Mueller. He also pointed us to an anomaly detector provided by Twitter. These tools help drive extraction of meaningful statistics and other characteristics of time series data in the timeline widget available in Neon GeoTemporal Dashboard (GTD). In this instance, we highlighted anomalous time buckets in red. The focused view component added to the timeline also helped alleviate scaling issues.

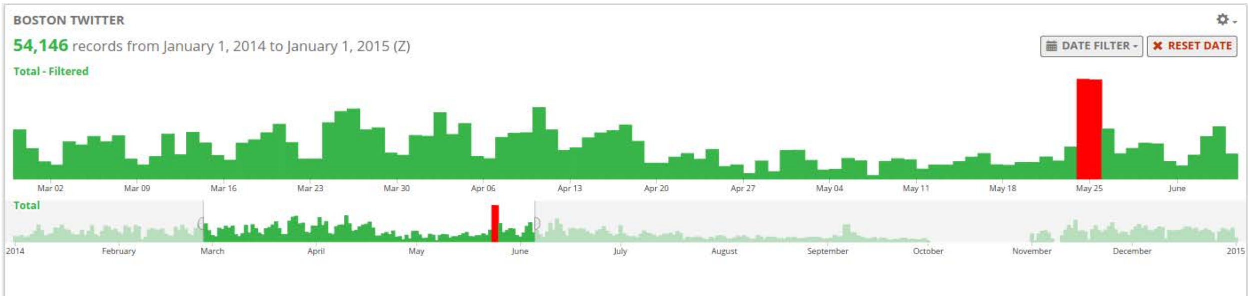

Figure 8: Time series refinement

Specific efforts at user-centered design incorporated into Neon were driven through several avenues. The Parsons Institute for Information Mapping, The New School provided assistance in improving the look and feel of existing visualizations. The widgets were redesigned based on Parsons' Design. The goal of these efforts was to generate a user-friendly interface in which disparate components nonetheless reflect a singular design ethos.

Similarly, we implemented a map layer solution using Uncharted's Aperture Tiles in the Neon Dashboard. This is a solution to an issue in which it is difficult to show map points in the example map application above a certain level, because all the data is sent to the client and rendered there. Using Uncharted's Aperture Tiles as a layer, our Dashboard allows offline rendering of the tiles. We also coordinated with Sotera to incorporate Louvain Modularity to be able to analyze taxi movement data over the map.

During the January 2016 Hackathon working with city permit data, Next Century teamed with two separate organizations to develop different visualizations of insights gleaned from that data. With CMU, we developed a dashboard presenting actual city permit valuations as well as their deviance from predicted valuations based on similar permits in similar areas. With MIT Lincoln Laboratory (MIT LL), we found and presented potential violations. MIT LL was able to explore their analytic products in Neon and present our collective work from within a Neon dashboard for the Outbrief presented to DARPA PM Wade Shen.

Also during the January 2016 Hackathon, the Next Century team was able to deliver specific, user-requested features. The first was the ability to import an XLS(X) or CSV file into the database with appropriate types. Another was a dashboard wizard that allowed selection of data sources and types, relationships, visualizations, and metrics. A third gave users the ability to save state at a URL location, offering them the ability to return to their work or share it with others. A fourth was alternate theming for the typically dark environment of SCIFs.

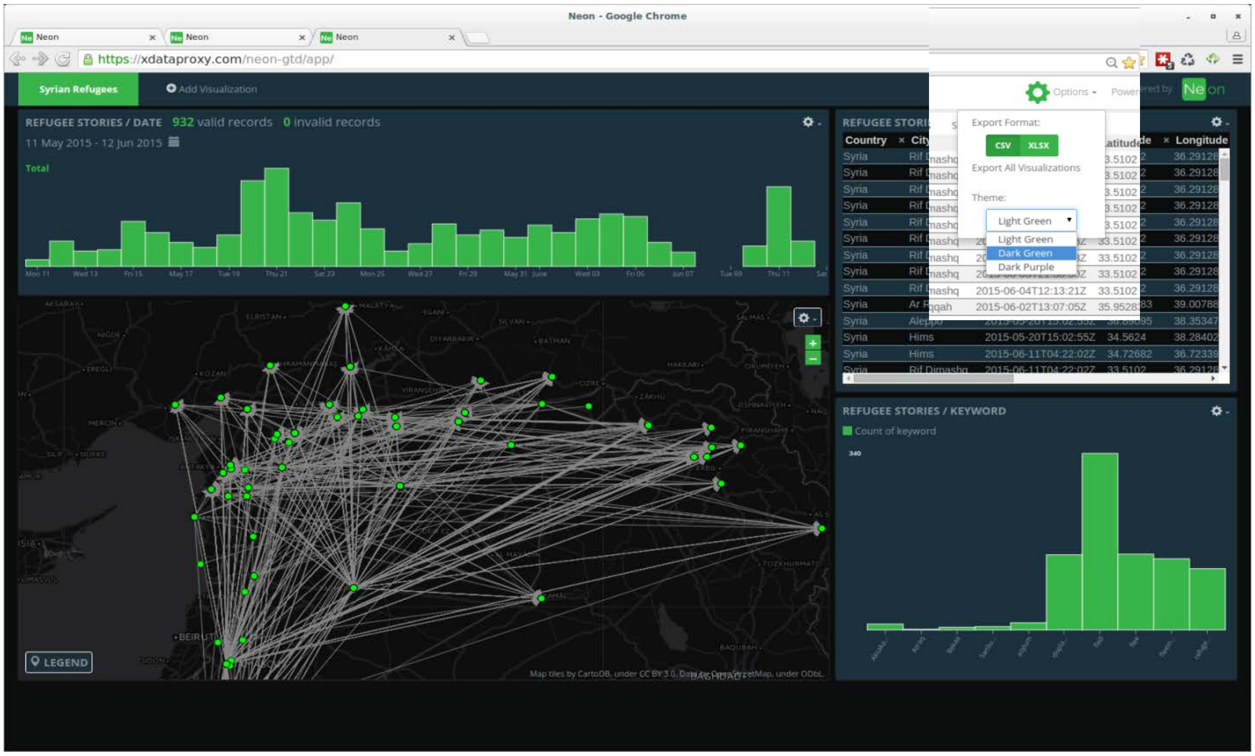

Figure 9 : Dark theme of Neon, with theme selector overlaid in the light theme

Finally, during the January 2016 Hackathon, the Next Century team worked to improve Core performance. We added ElasticSearch support to complement our ability to deal with MongoDB and SparkSQL data sources. In the process, we garnered an approximately 60% speed improvement just using the default caching scheme. However, working with ElasticSearch makes for more difficult ingest and indexing. The cache contains default dashboards and their queries and user-defined queries, but not trivial queries by time. Pre-computation is performed on startup, at admin's direction, or at some other external signal. In testing GTD performance, we saw on startup that there were more than 100 JS files, plus CSS files, icons, etc. We used concat, uglify, and sprite tools, as well as zipping five larger files to reduce load time by 30%. However, this slowed the system when it was run on networks.

4.1.2 SOCAF

In our pursuit of transition viability, we developed a demo version of the Neon interface to reflect Special Operations Command Africa (SOCAF) needs and workflow. This required establishing public (EC2) demo servers and additional Openstack servers in support of SOCAF work as demonstrated in this figure.

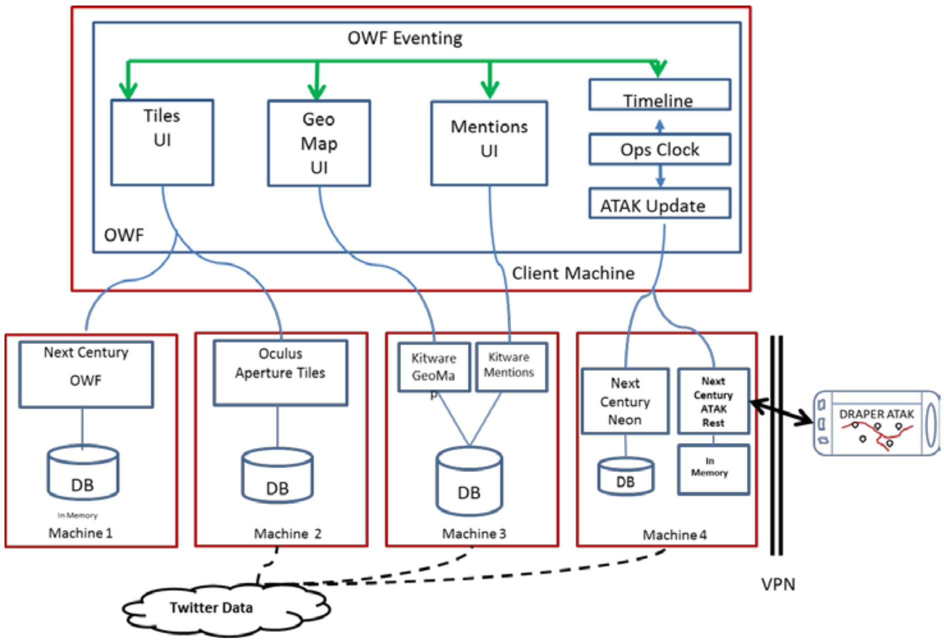

Figure 10: SOCAF architecture, 2014

Modifying Neon widgets allowed us to work with other SOCAF Dashboard implementers to send and receive appropriate information over OWF channels. Specifically, we coordinated filtering removal events. Deepening our relationship allowed us to attend the Information Systems Worldwide (ISW) presentation to SOCAF in August 2014 as well as pursue one-on-one discussions with Kitware, Uncharted, and Sotera to coordinate XDATA's programmatic efforts to support needs expressed by the Special Operations Command Africa team. Another portion of this work included a review of the STR-provided Twitter data, producing several datasets and covering single-day and multi-day analysis in Africa and Libya. Our strength in these presentations continued to be the inter-relationships among the various visualizations.

Figure 11: Neon cross-widget constraints

4.1.3 NSA ESpace Program Collaboration

Similarly, we built on existing Next Century relationships to work with the ESpace team at the NSA to create a Neon-integrated analyst widget for use with ESpace, using the same user interface framework as is already familiar to analysts working in that software. We were able to integrate with existing ESpace and Synapse widgets using ESpace ResultSet tables. This provided a new 15-day view widget that may point at a new data source. As part of this effort, we prototyped an integrated ESpace/Neon widget that would more closely align with the 15-day view widget by pulling in the ESpace map into the Neon demo application and using Neon components to display visual data from an ESpace result set. Additionally, we used the Neon temporal and spatial filters to adjust the information displayed in the ESpace map. We demonstrated this to ESpace Technical Director, Sheldon Bateman as a working ESpace dashboard application in February 2015. We provided technical information to the ESpace team as that government customer was attempting to stand up an operational analysis dashboard based on Neon and the Neon GTD, helping transition Neon capabilities to the operational environment.

In the screen capture images included here, we demonstrate the core Neon capacity of driving interactions across widgets, such that map bounds limit Neon aggregations in one direction, or the time ranges pass from Neon to the ESpace map in the other. Additionally, the OpsClock, Map, and Timeline all work together, in either point mode or heatmap mode, as demonstrated in the two screen capture views included here in Figure 13.

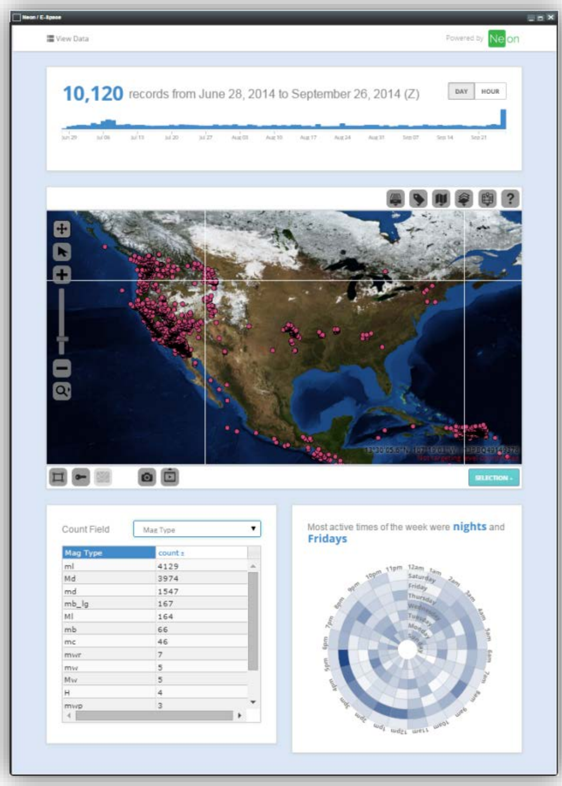

Figure 12: ESpace demo of Neon

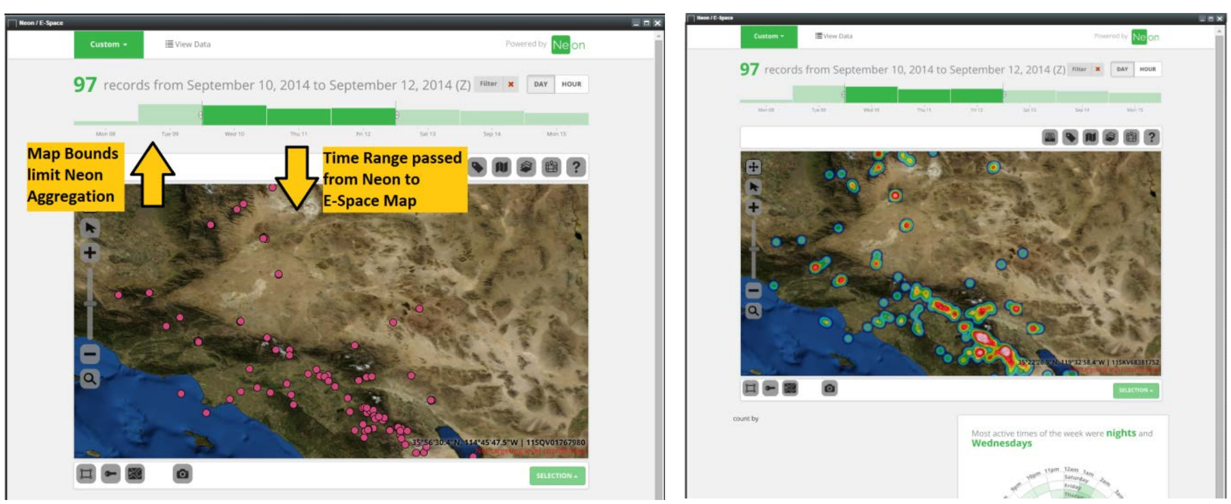

Figure 13: ESpace integration with Neon in two screen capture views.

4.2 Pentagon DARPA Demo Days

The Neon team participated in two Pentagon DARPA Demo Days in May 2014 and 2015, both to share findings and to generate interest in the military and intelligence user communities. Our ability to present responsive widgets using the Neon framework was based on work preparing a Twitter dataset for showing, as well as code and testing support. These opportunities allowed us to refine Neon demo components based on feedback received over the course of the events. The 2014 presentation still featured the Ozone Widget Framework basis for enabling technology-agnostic composition of web browser-based applications in a common display environment, combined with the basic Neon Framework (Server, Client, and Widgets) as described in Section 3.1. By the time of the 2015 Demo Day, the system had evolved to include the Neon GeoTemporal Dashboard overlaid on the Neon Core Project to allow for dashboard filtering and visualization integrations as demonstrated in the following figures.

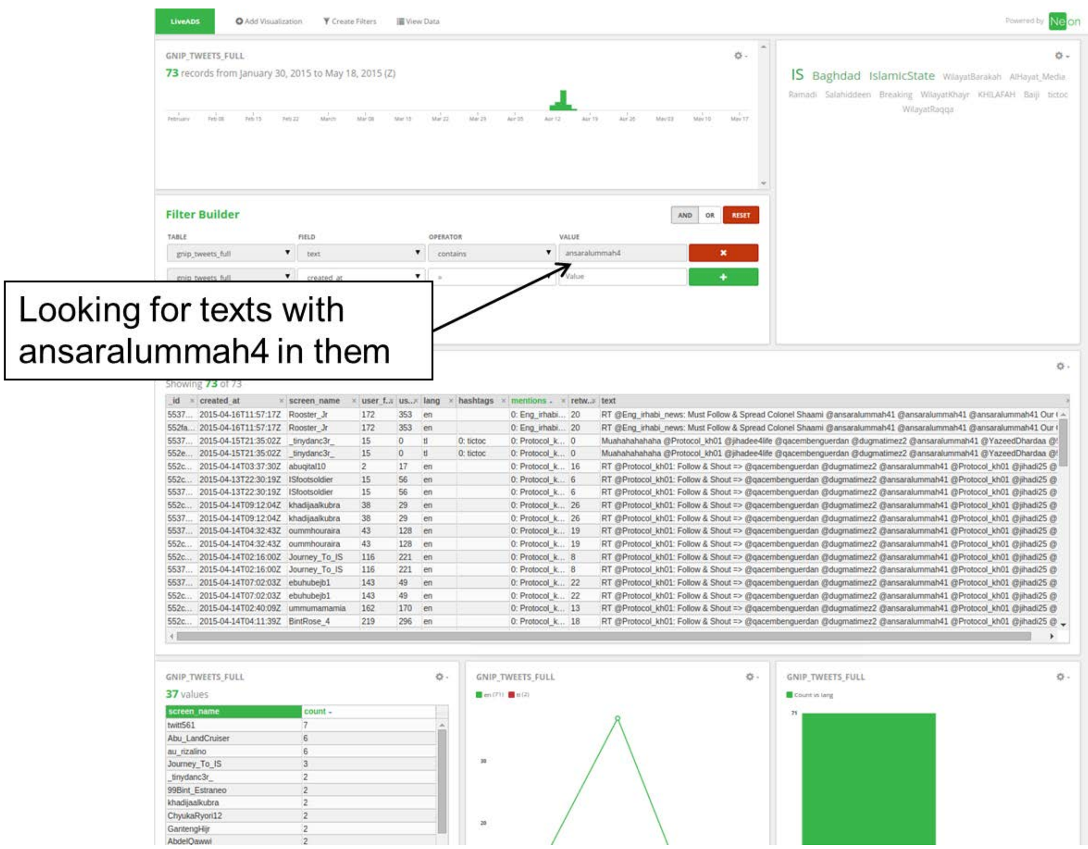

Figure 14: Neon Search filter builder

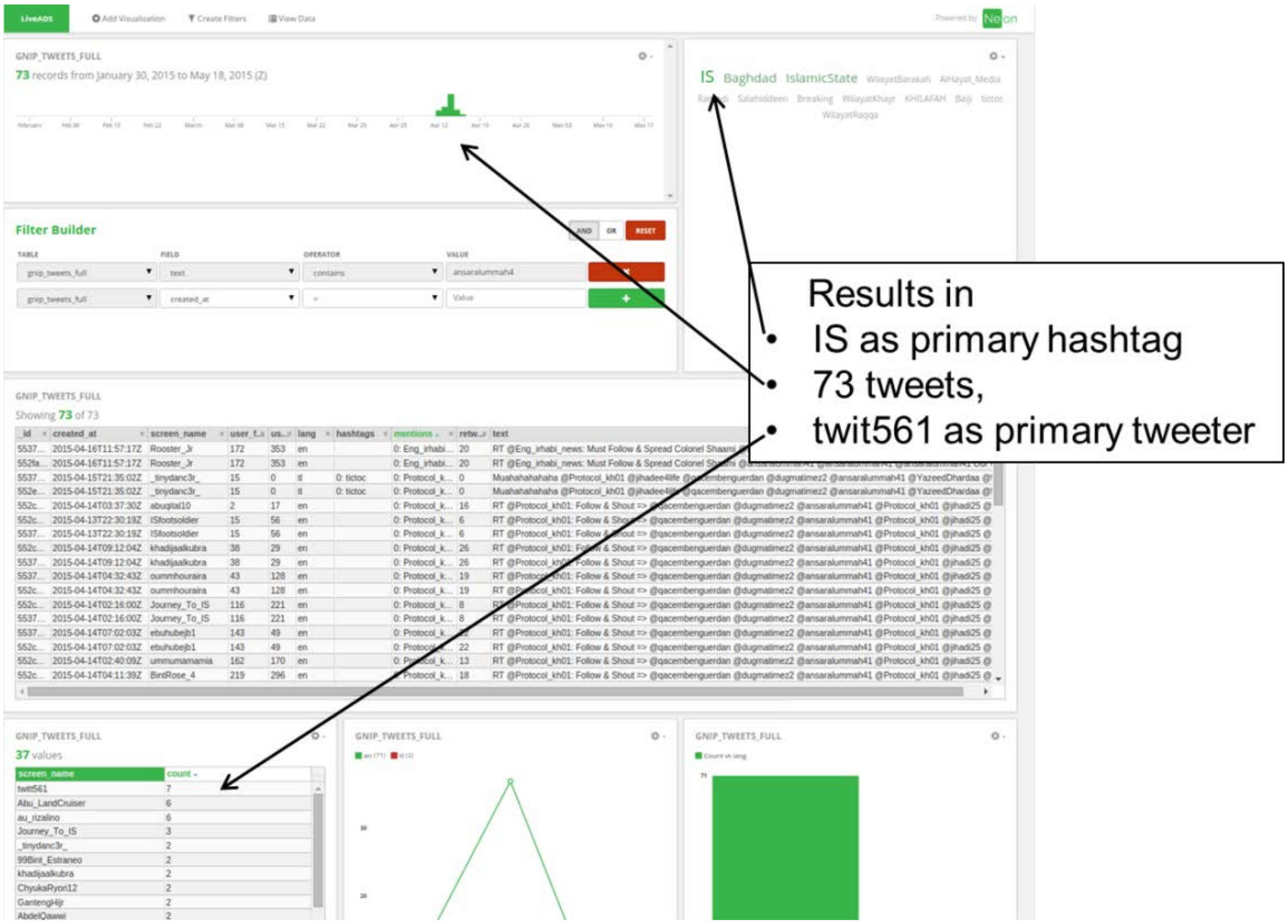

Figure 15: Filter results

4.3 DARPA'S Quantitative Crisis Response (QCR)

Quantitative Crisis Response (QCR) was a subproject within the XDATA program that explored data analysis for burgeoning crises. It developed suites of largely automated digital tools that can help operational partners better understand how information is being used by adversaries and to quantitatively predict and assess—in real time and at scale—the effects of those campaigns and of countermeasures. We participated in both unclassified and classified analyses under the QCR rubric. On the QCR effort, we filled the role of team integrator, providing the initial dashboard for event datasets. In addition to leveraging Neon for first-pass analysis, Neon linked to other tools in a context appropriate manner. The Neon dashboard served as the launching point for those specialized data analysis tools.

This effort focused on the ability to port Neon tools high-side. To accomplish this, we worked with Kitware to demonstrate Tangelo Mentions running alongside Neon GTD visualizations. We also worked with Data Tactics to deploy and configure Neon, demonstrating high-side capability by visualizing detainee data as part of the 2015 Summer Hackathon exercise. We used the D3-based network to show friends and followers in Twitter data and worked with Glenn Coppersmith, a researcher with Johns Hopkins University to develop annotations for Tweets. We added a feedback mechanism for analysts to evaluate Tweets' sentiment. Mr. Coppersmith received propaganda data and analyzed it, working with us to develop a dashboard to visualize propagation of propaganda through social networks. In addition to developing various visualizations, (e.g., Venn Clouds, https://github.com/coppersmith/vennclouds), he performed a lot of the data collection and parsing on core datasets used within XDATA. Working with other team members, we obtained movement data for Syrian refugees and camps and added a new map layer that shows directions of movement. Additionally, working with Hyperion Gray, we were able to improve links between applications.

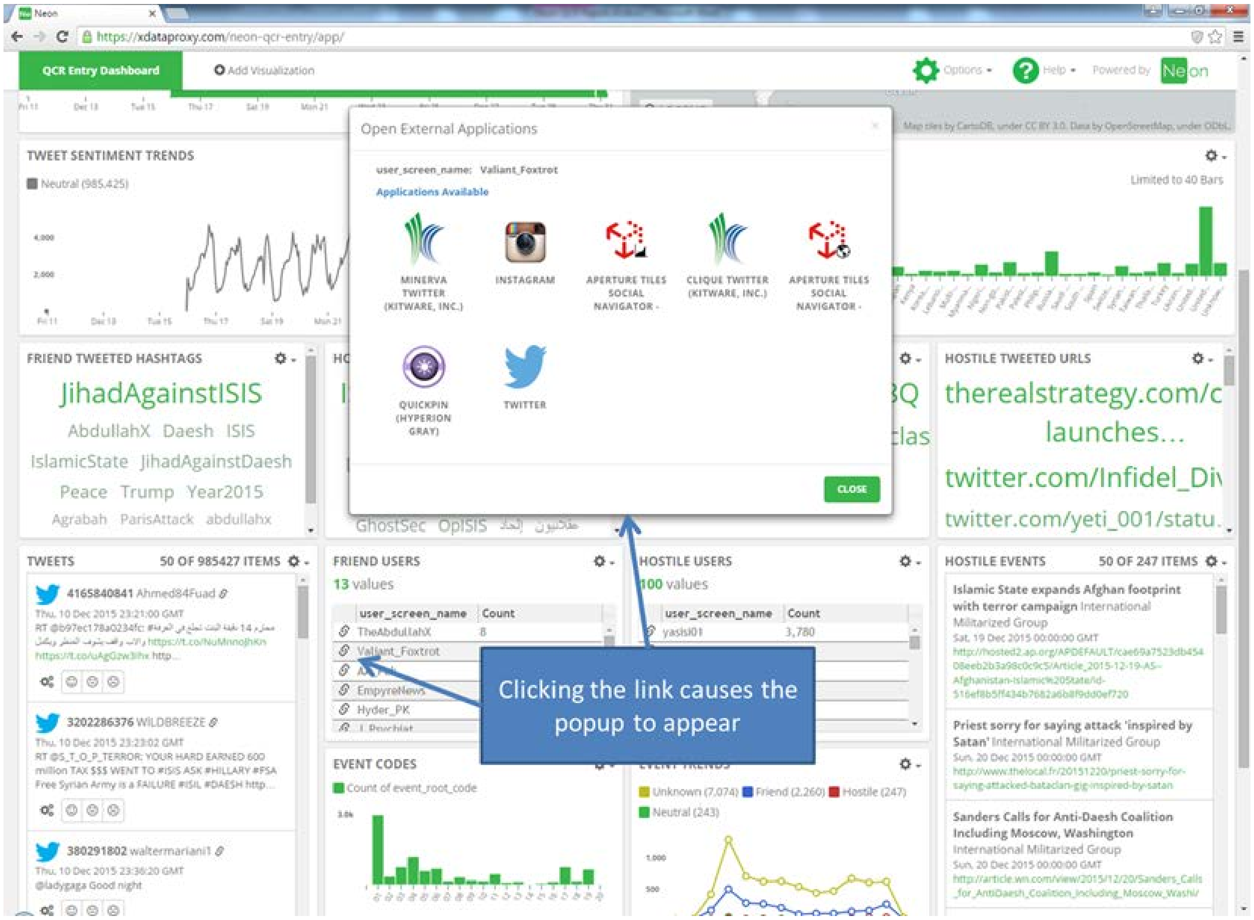

Figure 16: QCR dashboard

Our work culminated in two additional opportunities for funding. We presented Neon at the Defense Intelligence Information Enterprise (DI2E) Plugfest. (http://di2eplugfest.org) on June 1&2, 2016 as part of our early marketing efforts. We also presented a proposal for a Neon-based Cyber Analysis dashboard to an Army sponsor. The proposal made the first down-select and the government has expressed interest in establishing a contractual relationship based on this work. Discussions are still ongoing.

4.4 Scalability Testing

Dynamic aggregations are fundamentally hard for any system. To explore which might provide the most robust solution for Neon users, the Next Century team built off a review of the open source landscape of data providers, including Shark, Apache Gora, OODT, Solr projects, HDFS-enabled databases, MongoDB, CouchDB, Cassandra, and others. ElasticSearch and Solr operate on very different query models, while Impala and Presto.io are comparable to Spark in other benchmarks.

Nanocubes are designed for aggregations, but work in opposition to the framework our team established. The evaluation was intended to review database technologies to improve scalability. It helps handle the end-of-life of Shark, the scalable query engine on top of Spark, which is replaced by Spark SQL. Using Amazon EC2 machines, we compared scalability specifically between MongoDB, Spark SQL, and Postgres.

The results from a single machine showed that MySQL and PostgreSQL took 5-15 seconds for aggregation queries, while Spark was slightly faster (2-10 seconds), and MongoDB was slightly slower (10-20 seconds). For all except Spark, small subsets took around 0.01 seconds; Spark takes roughly the same amount of time for all queries, regardless of size. With parallelization, MongoDB and Spark experience roughly linear speed ups. The caveats with that are an initial performance hit with the first few nodes, and Spark cannot go faster than roughly 0.5 seconds.

4.5 Widgets

The Neon team was able to successfully develop eighteen specific widgets for use in the Neon GeoTemporal Dashboard (GTD) web app interface. These are: timeline; bar chart; line chart; map; heat map; points; lines and arrows; filter builder; ops clock; table; force-directed graph; count-by widget; sunburst; and scatterplot. Each serves specific data analyst needs and has been vetted against real user needs to ensure the product addresses real-world use cases as opposed to merely representing an engineering conundrum. Examples follow in images screen captured from the Neon GTD interface:

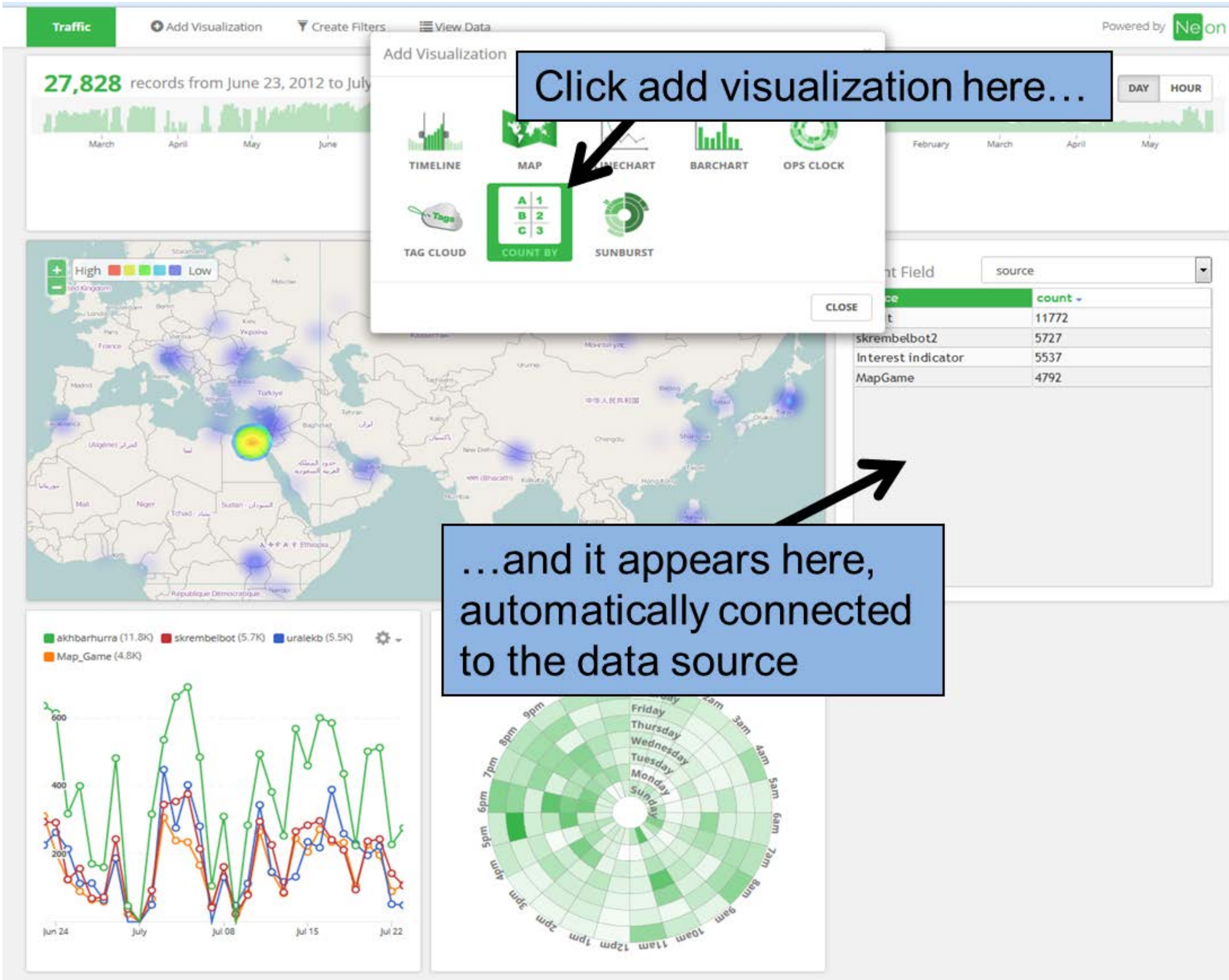

Figure 17: Neon widget selection tool screen capture

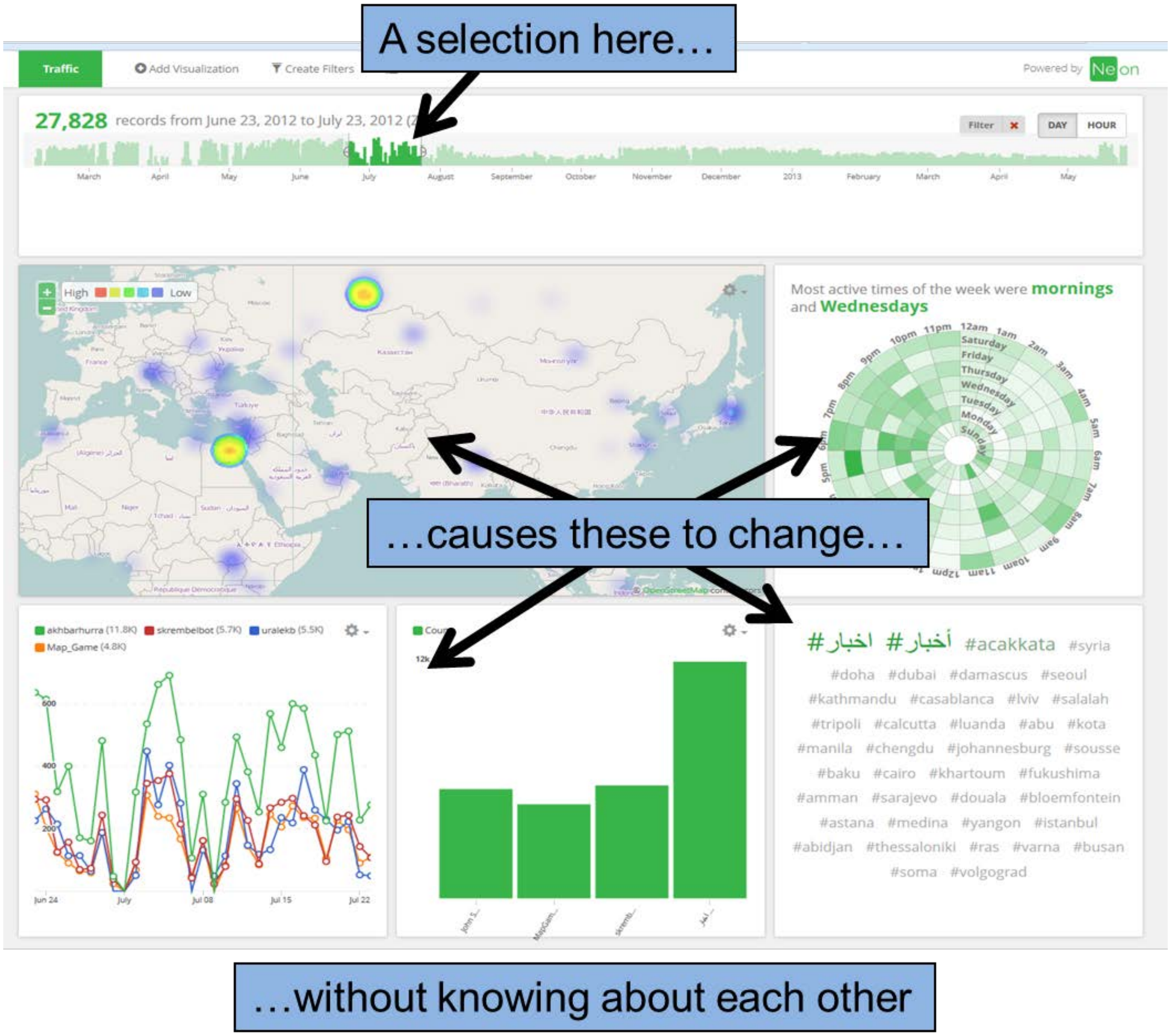

Figure 18: Linked data displayed across Neon widgets

5. CONCLUSIONS

There is great promise in the Neon toolset developed under the XDATA contract. It has already grown beyond its original concept and found uptake in additional environments and circumstances, and will continue its development lifecycle in diverse projects because the ability to visualize data is such a widespread need. By having made the Neon tools an ecosystem based on two powerful APIs, Next Century has helped further the XDATA goal of effectively scaling to the volume and characteristics of changing data environments and providing tools that allow for wide-ranging data analysis. In particular, interactive visualization that goes beyond old-style stovepipe concepts means analysts can now connect their databases in an agnostic way to tools that will simplify their research tasks and drive their ability to deepen their interaction with their datasets. The Neon Core contains a Data Access API that allows users to send queries to NoSQL databases using SQL-like language. Neon converts the query to a format understood by the target database, removing the need for developers to create database-specific constructs. It also contains an Interaction API to control inter-widget communication, allowing developers to orchestrate complex, interactive visualizations using components that are completely decoupled from each other. On the front end, Neon GTD visualizes the data the two APIs link it to, providing a unified view of that data across multiple visualization options. These visualizations and interactions have been proven and improved upon via collaborations with other XDATA performers, as well as Draper Labs' user tests.

Draper Labs used its SensSoft tool in October 2016 to A/B test layouts, establish simple statistics about correlations between frequentist metrics and outcome measures, perform a widget workflow analysis, and graph an activity flow analysis. In lieu of a formal evaluation team, having Draper validate our approach by being able to construct tests in our framework and both validate their hypotheses as well as provide specific usability feedback for later iterations of the default setup for the GTD dashboard, where they had not been successful in doing so for the other visualization teams, meant Next Century's solution effectively achieved its goal of having its widgets rate highest among performers.

The two other goals outlined at the beginning of the program, having a median development time that was ten times faster than conventional programming and setup requires, as well as the ability to have 25 simultaneous users on a single widget dashboard, were not as rigorously tested. However, as a proof of concept, having been able to add at least one new visualization per Hackathon marked a significant speed improvement for that kind of integration. Similarly, by architecting our solution to allow for a saved URL to share among team members, we proved at least a baseline level of collaborative capacity.

5.1 Follow-On Development under Additional Contracts and SBIRs

The team's work on visualization dashboards is compelling for a number of additional contracts and work under development. Five DARPA programs benefit from this cross-fertilization. Specifically, starting with VINI and visualizations being developed for the ADAMS and SMISC programs at DARPA, we exchange code and development philosophies. We assisted in creating a demo using Neon and VINI STTR tools as a visualization mechanism for MUSE. With Memex, we developed a Neon-based Memex analysis dashboard. The Next Century LORELEI project is furthering the work begun under XDATA, with its user interface based on Neon, refining lessons learned from subject matter experts relevant to that program. The prototype LORELEI interface has already made the leap into use in the NSA. This allows for continuous cross-fertilization between the programs, incorporating both teams' Neon GTD enhancements in the primary branch of its development. Specific widgets that resulted from this collaboration include a scatterplot and a 2D sentiment visualization. Similarly, a collaboration with RSPACE program developers allows us to assist in the development of the architecture and to provide input for how to use Neon and other XDATA-developed tools in work that will ultimately serve the Air Force.

Other government agencies that have seen examples of how Neon could be transitioned to meet their needs include the US Navy and the NGA. Developers responding to an SBIR for the Navy used Neon and the Neon Example Application to show influenza data. Another group provided a demonstration of Neon to NGA clients as part of ongoing support conversations. In both cases the demos were positively received, though no additional funding was forthcoming.

5.2 Commercialization

We commit to promote the transition of Neon: Next Century will employ our *continuous delivery* software development process to release regular builds of Neon for use by the XDATA developer community; deliver Neon with Unlimited Rights in its entirety, so intellectual property will not impede adoption; build on open source software widely recognized across the DoD and IC, so that Neon may inherit a community of thousands of users and developers; make use of feedback from Neon early adopters to evolve the system; fold mature elements of Neon developed under XDATA into future releases of the OWF baseline; and release Neon as open source software (with DARPA approval) so that others may benefit from it and contribute to its long-term sustainment.

In addition, we developed and maintained the Neon website (at http://www.neonframework.org/) to publicize new releases and make the links to our code repositories widely available. There is currently an ad campaign underway to promote developer participation in and awareness of our GitHub repository; it will run through early 2018 to extend the life of potential interest in the tools developed under this program.

KEY PERSONNEL

Principal Investigator: Mr. Michael Tamayo, Next Century Corporation

Mr. Tamayo has twenty years of experience applying advanced computer science techniques to large-scale programs. He served as technical lead on DCHIP, a large-scale enterprise portal based on OWF that serves as the Human Intelligence hub of the Army's Distributed Common Ground System (DCGS-A). He also served as technical lead on the Threat Warning System ACTD for U.S. Special Operations Command, which implemented a body-worn radio-frequency direction-finding system that provides intuitive sensor information visualization on a mobile device. Prior to joining Next Century, Mr. Tamayo led the development of software products that deliver Web content designed for desktop computers to mobile devices. He also designed distributed software architectures that remained operational when deployed in an environment with intermittent network connectivity. Mr. Tamayo earned a B.S. in Computer Science from Florida State University.

REFERENCES

K. Alexandros and H. Melanie. Model selection via metalearning: a comparative study. International Journal on artificial intelligence tools 10 (4), 2001.

Apache Wave, http://incubator.apache.org/wave/.

A-Y.D. Chiang, D.J. Berg, L. Itti. Saliency, Memory, and Attention Capture in Marketing. In Proceedings of Vision Science Society Annual Meeting (VSS11), 2011.

CodeNarc, http://codenarc.sourceforge.net/.

L. Costello, G. Grinstein, C. Plaisant, J. Scholtz. Advancing User-Centered Evaluation of Visual Analytic Environments through Contests. Information Visualization 8, 2009.

Cucumber, http://cukes.info/.

Data-Driven Documents, http://d3js.org/.

D.J. Duke, K.W. Brodlie, D.A. Duce. Building an ontology of visualization. In Proceedings of the conference on Visualization, 2004.

K.A. Ehinger, B. Hidalgo-Sotelo, A. Torralba, A. Oliva. Modelling search for people in 900 scenes: A combined source model of eye guidance. Visual Cognition 17, 2009.

C. Eick, A. Rouhana, A. Bagherjeiran, R. Vilalta. Using Clustering to Learn Distance Functions for Supervised Similarity Assessment. MLDM, 2005.

C.A. Ellis and S.J. Gibbs. Concurrency control in groupware systems. ACM SIGMOD Record 18 (2), 1989.

D.B. Hellar and L. Vega. The Ozone Widget Framework: Towards Modularity of C2 Human Interfaces. Proceedings of SPIE: Defense, Security + Sensing, 2012.

L. Itti, C. Koch, E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11), 1998.

JavaScript InfoVis Toolkit, http://thejit.org/.

Jenkins, http://jenkins-ci.org/.

JUnit, http://www.junit.org/.

C. Köpf, C. Taylor, J. Keller. Meta-analysis: From Data Characterisation for Meta-learning to Meta-regression. In Proceedings of the PKDD-2000 Workshop on Data Mining, Decision Support, Metalearning and ILP: Forum for Practical Problem Presentation and Prospective Solutions, 2000.

Z. Li and L. Itti. Saliency and Gist Features for Target Detection in Satellite Images. IEEE Transactions on Image Processing 20 (7), 2011.

M.L. LarreA, S.R. Martig, S.M. Castro. Semantics-based color assignment in visualization. JCS&T 10 (1), 2010.

G. Leban, B. Zupan, G. Vidmar, I. Bratko. VizRank: Data Visualization Guided by Machine Learning. Data Mining and Knowledge Discovery, 13 (2), 2006.

D. Li and R. Li. An Admissibility-Based Operational Transformation Framework for Collaborative Editing Systems. 19, 2010.

D. Parkhurst, K. Law, E. Niebur. Modeling the role of salience in the allocation of overt visual attention. Vision Research 42, 2002.

G.P. Oster, P. Urso, P. Molli, A. Imine. Real Time Group Editors without Operational Transformation. INRIA Technical Report, 2005.

PhiloGL, http://www.senchalabs.org/philogl/.

M. Ressel, D. Nitsche-Ruhland, R. Gunzenhäuser. An integrating, transformation-oriented approach to concurrency control and undo in group editors. In Proceedings of the 1996 ACM conference on Computer supported cooperative work, 1996.

R. Spence. Information Visualization - Design for Interaction (2nd Edition). Pearson Education, 2006.

D. Sun and C. Sun. Context-based Operational Transformation in Distributed Collaborative Editing Systems. IEEE Transactions on Parallel and Distributed Systems 20 (10), 2009.

VAST Challenge 2012, http://www.vacommunity.org/VAST+Challenge+2012/.

M. Verma, P.W. McOwan. A semi-automated approach to balancing of bottom-up salience for predicting change detection performance. Journal of Vision 10 (6), 2010.

I. Vessey. Cognitive Fit: A theory-based analysis of the graphs versus tables literature. Decision Sciences 22 (2), 1991.

R. Vilalta. Identifying and Characterizing Class Clusters to Explain Learning Performance. AAAI Spring Symposium: What Went Wrong and Why: Lessons from AI Research and Applications, 2006.

R. Vilalta, C.G. Giraud-Carrier, P. Brazdil. Meta-Learning. The Data Mining and Knowledge Discovery Handbook, 2005.

R. C. Voorhies, L. Elazary, L. Itti. Application of a Bottom-Up Visual Surprise Model for Event Detection in Dynamic Natural Scenes. In Proceedings of Vision Science Society Annual Meeting (VSS10), 2010.

VTK: The Visualization Toolkit, http://www.vtk.org/.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ACTD – Advanced Concept Technology Demonstrations

ADAMS – Anomaly Detection at Multiple Scales, a DARPA program that creates, adapts and applies technology to anomaly characterization and detection in massive data sets.

API – Application program interface

ATAK – Android Terminal Assault Kit

CMU – Carnegie Mellon University

CSV – Comma-Separated Value

DARPA – Defense Advanced Research Projects Agency

DCGS-A – Army's Distributed Common Ground System

DCHIP - Deployable Counterintelligence/Human Intelligence Portal

DoD – Department of Defense

EC2 – Amazon's Elastic Compute Cloud

ESpace – NSA program of record

G-OLA – Generalized Online Aggregation

HDFS – Hadoop Distributed File System

HTTP– HyperText Transfer Protocol is a way to transfer files.

IC – Intelligence Community

LORELEI – Low Resource Languages for Emergent Incidents, a DARPA program that aimed to dramatically advance the state of computational linguistics and human language technology to enable rapid, low-cost development of capabilities for low-resource languages.

Memex – DARPA program launched to advance the online search capabilities far beyond the current state of the art, including domain-specific technologies to revolutionize the discovery, organization, and presentation of domain-specific content.

MIT – Massachusetts Institute of Technology

MUSE – Mining and Understanding Software Enclaves, a DARPA program that sought to make significant advances in the way software is built, debugged, verified, maintained and understood.

NSA – National Security Agency

OODT – Object Oriented Data Technology is an open source data management system framework that is managed by the Apache Software Foundation.

OWF – Ozone Widget Framework, originally developed by Next Century, currently a program of record in the IC.

QCR – Quantitative Crisis Response, a DARPA program designed to rigorously assess the effects of the volleys of information that are traded through social media and other communications channels.

PIIM – Parsons Institute for Information Mapping

REST – Representational State Transfer is an architectural style for designing distributed systems.

RSPACE – Resilient Synchronized Planning and Assessment for the Contested Environment, a DARPA program that seeks to create a revolutionary distributed planning capability to provide resilient command and control (C2) and to manage complex military operations even when communications are limited and unreliable.

SBIR – Small Business Innovation Research, a highly competitive program that encourages domestic small businesses to engage in Federal Research/Research and Development.

SMISC – Social Media in Strategic Communication, a DARPA program that seeks to develop a new science of social networks built on an emerging technology base.

SOCAF – Special Operations Command Africa

STR – Systems & Technology Research, organization - http://www.stresearch.com/

VINI – Visual Interaction for Network Information, a DARPA program.

STTR – Small Business Technology Transfer, a program established by congress in 1992.